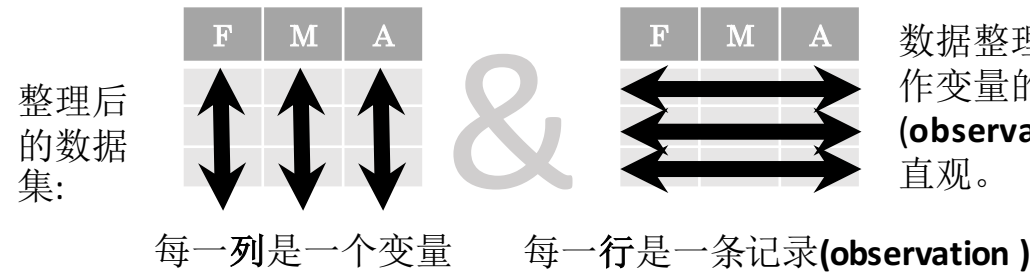


Data Wrangling

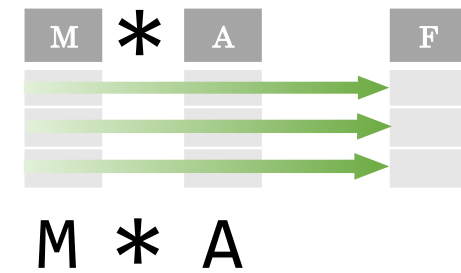
with pandas Cheat Sheet

<http://pandas.pydata.org>

数据整理 - pandas数据处理的基础



数据整理辅助了pandas的向量操作. 当你操作变量的时候, pandas将自动保存记录(observation), 没有其他格式能比pandas更直观。



语法 - 创建DataFrame

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
为每一列指定值
```

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
为每一行指定值
```

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

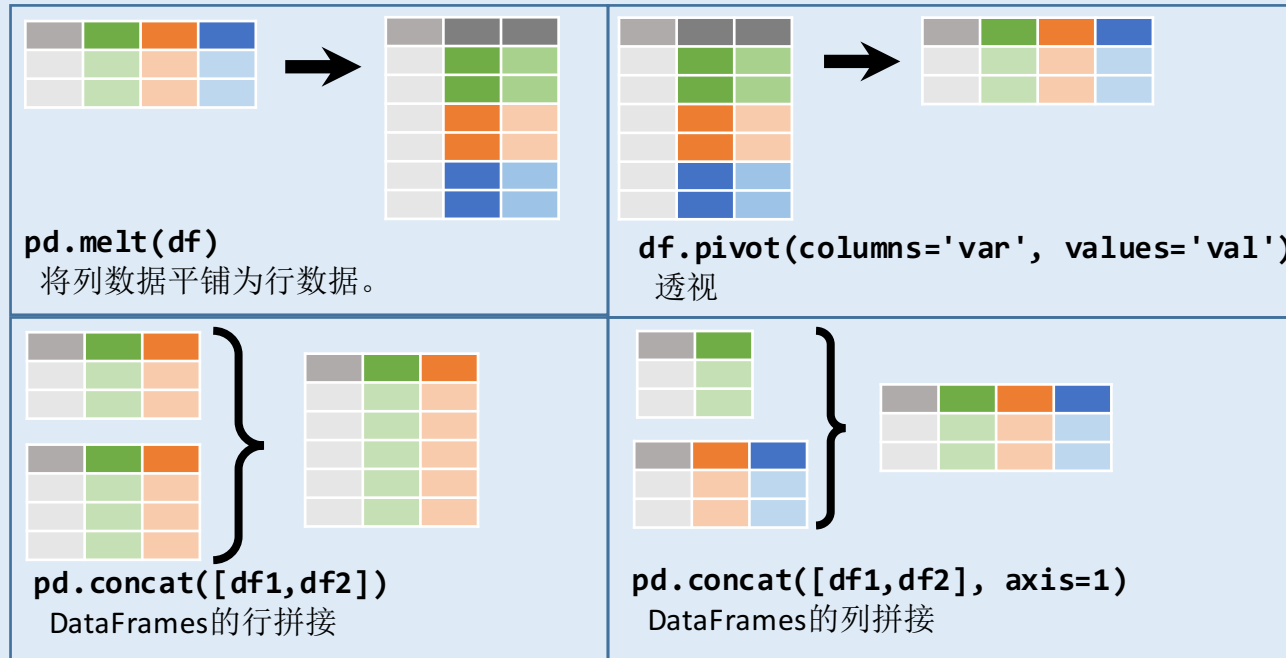
```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d', 1), ('d', 2), ('e', 2)],
        names=['n', 'v']))
创建多索引(MultiIndex)的DataFrame
```

链式调用

大多数pandas方法返回一个DataFrame对象, 以便可以链式调用另一个方法, 这提高了代码的可读性。

```
df = (pd.melt(df)
      .rename(columns={
          'variable': 'var',
          'value': 'val'})
      .query('val >= 200'))
```

数据重塑 - 改变数据集的布局



```
df.sort_values('mpg')
按照某列的值进行行排序(升序)

df.sort_values('mpg', ascending=False)
按照某列的值进行行排序(降序)

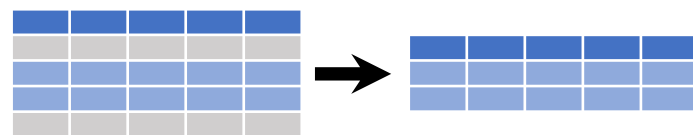
df.rename(columns = {'y': 'year'})
列名重命名

df.sort_index()
按索引排序

df.reset_index()
重置DataFrame的索引为行号, 原索引变为普通列

df.drop(columns=['Length', 'Height'])
移除特定列
```

子集(行)



```
df[df.Length > 7] 筛选满足逻辑表达式的数据
df.drop_duplicates() 移除重复行 (仅考虑列)
df.head(n) 选择前n行
df.tail(n) 选择后n行

df.sample(frac=0.5) 随机选取指定比例的行
df.sample(n=10) 随机选取n行
df.iloc[10:20] 按位置选取行
df.nlargest(n, 'value') 按列值排序选取前n行
df.nsmallest(n, 'value') 按列值排序选取后n行
```

子集(列)



```
df[['width', 'length', 'species']] 选取指定的多列
df['width'] or df.width 选取指定的单列
df.filter(regex='regex') 选取列名满足正则表达式的列
```

正则表达式示例

'\.'	包含符号'
'Length\$'	以字符串'Length'结尾
'^Sepal'	以字符串'Sepal'开头
'^x[1-5]\$'	以字符串'x'开头且以字符串'1'或'2'或'3'或'4'或'5'结尾
'^(?!Species\$).*\$'	除'Species'外的字符串

```
df.loc[:, 'x2': 'x4'] 选取x2(含)和x4(含)之间的列
df.iloc[:, [1, 2, 5]] 选取下标为1,2,5的列(第一列下标为0)
df.loc[df['a'] > 10, ['a', 'c']] 选取满足逻辑表达式的行, 但只返回指定列'a', 'b'
```

Python(pandas)中的逻辑运算			
<	小于	!=	不等于
>	大于	df.column.isin(values)	在...内, 属于关系
==	等于	pd.isnull(obj)	值为null
<=	小于等于	pd.notnull(obj)	值非null
>=	大于等于	&, , ~, ^, df.any(), df.all()	逻辑与, 或, 非, 异或, 存在, 全部

数据统计

df['w'].value_counts()
统计w列每一个值出现的次数，并降序排列

len(df)
返回DataFrame的行数

df['w'].nunique()
返回w列去重后值的个数

df.describe()
数据值列描述性汇总统计信息 (GroupBy 对象也有该方法)



pandas提供了大量的汇总函数 (summary functions)，它们对不同种类的pandas对象 (DataFrame列, Series, GroupBy, Expanding和Rolling (见下文)) 进行操作，并为每个group生成单个值。作用于DataFrame时，返回结果是Series对象。

sum() 求和

count() 统计non-NA/null值的个数

median() 计算中位数

quantile([0.25,0.75]) 计算p分位数的值

apply(function) 将传入函数作用于每个对象

min() 求最小值

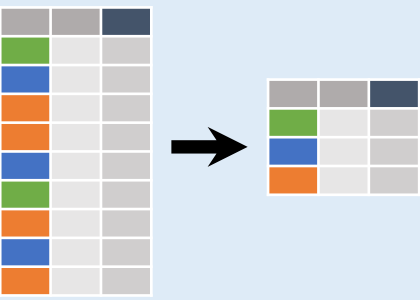
max() 求最大值

mean() 求平均值

var() 求方差

std() 求标准差

数据分组



df.groupby(by="col")
返回GroupBy对象
按'col'列的值进行分组

df.groupby(level="ind")
返回GroupBy对象
按'ind'的index进行分组
常用于MultiIndex场景下

上文所述的所有的汇总函数(summary functions)都能用于group
其他的GroupBy对象函数:

size() 每个group的大小

agg(function) 使用function对group进行聚合

窗口

df.expanding()
返回一个Expanding对象，在该窗口内，可以累计作用汇总函数(summary functions)

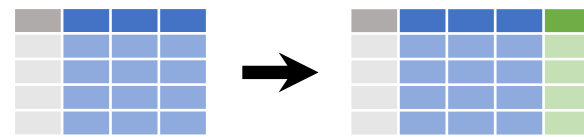
df.rolling(n)
返回一个Rolling对象，在长度为n的滚动窗口内，作用汇总函数。

处理缺失值

df.dropna()
去掉含有NA值的行

df.fillna(value)
将所有的NA/null值替换为value

添加新列



df.assign(Area=lambda df: df.Length*df.Height)
计算并添加一列或多列

df['Volume'] = df.Length*df.Height*df.Depth
添加一列

pd.qcut(df.col, n, labels=False)
根据值的频率将数据分为n组，返回值为分组下标



pandas提供了大量的向量函数 (vector functions)，可以作用于DataFrame的所有列或单个选定列 (Series对象)。这些函数为每个列生成值向量。例子:

max(axis=1) 求每行最大值(axis可指定)

min(axis=1) 求最小值

clip(lower=-10,upper=10) 将边界值外的值置为边界值

abs() 求绝对值

下面的例子也能用于group，在这种场景下，这些方法作用于group的每一项，返回的向量长度与原DataFrame一致。

shift(1) 对数据进行移位

rank(method='dense') 排名，同值同排名，紧凑排名

rank(method='min') 排名，同值同排名，每次增加1

rank(pct=True) 排名，结果映射到[0, 1]

rank(method='first') 排名，值相同时按出现顺序排

shift(-1) 对数据进行反向移位

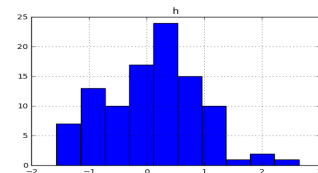
cumsum() 累加

cummax() 累计最大值

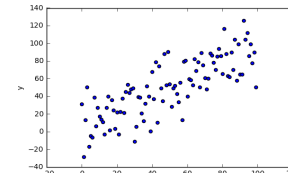
cummin() 累计最小值

cumprod() 累乘

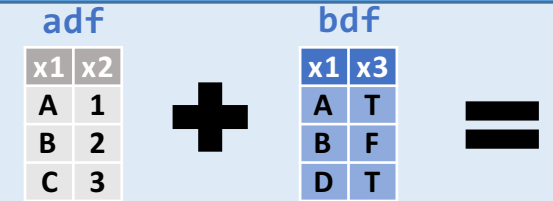
df.plot.hist()
直方图(按列)



df.plot.scatter(x='w',y='h')
散点图



合并数据集



Standard Joins

pd.merge(adf, bdf, how='left', on='x1')
bdf与adf合并，以adf为基准

pd.merge(adf, bdf, how='right', on='x1')
adf与bdf合并，以bdf为基准

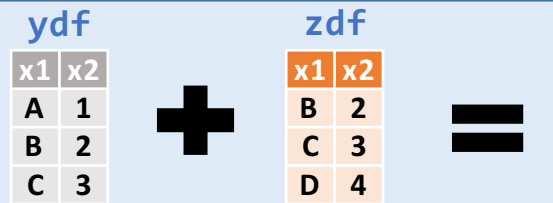
pd.merge(adf, bdf, how='inner', on='x1')
合并数据，保留x1列的值均存在的行

pd.merge(adf, bdf, how='outer', on='x1')
合并数据，保留所有行，所有数据

Filtering Joins

adf[adf.x1.isin(bdf.x1)]
返回adf满足过滤条件的行(x1在bdf.x1存在)

adf[~adf.x1.isin(bdf.x1)]
返回adf满足过滤条件的行(x1在bdf.x1不存在)



Set-like Operations

pd.merge(ydf, zdf)
同时出现在ydf和zdf的行(交集)

pd.merge(ydf, zdf, how='outer')
出现在ydf或zdf的行(并集)

pd.merge(ydf, zdf, how='outer', indicator=True).query('_merge == "left_only").drop(columns=['_merge'])
在ydf出现但不在zdf出现的行(差集)

绘图